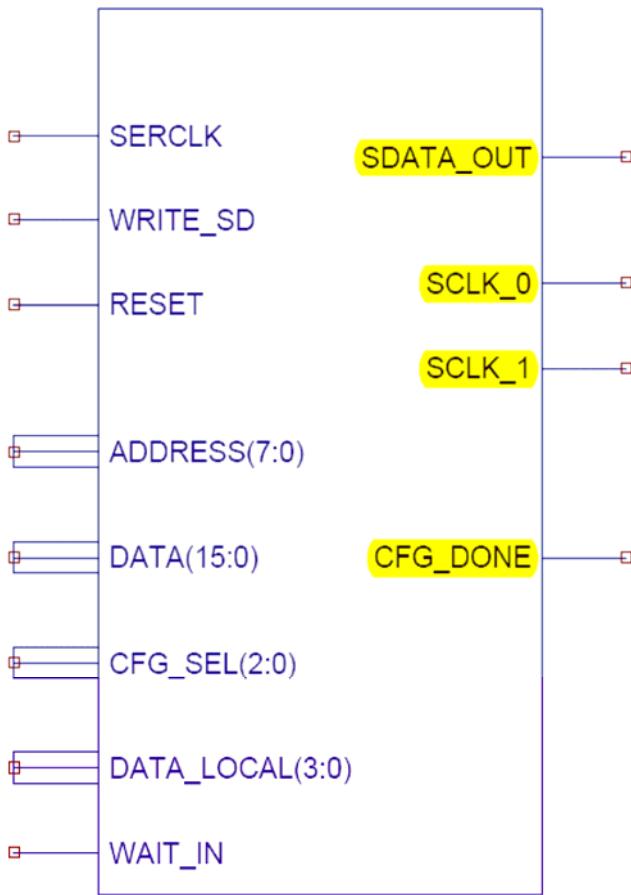


4/5/2006

```
-- DESCRIPTION:  
-- This module provides the serial data stream for the configuration  
-- devices on the CPLDs. This stream is assembled out of the parallel  
-- data coming directly from the backplane interface (Video ADC cfg,  
-- CDS cfg, DAC cfg and Auxiliary functions) or from the cfg_mem_module  
-- (Tel ADC cfg), plus the address information from the backplane  
-- interface.  
--  
-- The stream format is:  
-- | START | FUNCTION | DEV ADDR | MSG LENGTH | MESSAGE | STOP |  
-- | 1bit | 3 bits | 6 bits | 6 bits | MSG LENGTH bits | 1bit |  
--  
-- NOTES:  
-- SDATA IS SENT MSB FIRST.  
-- START BIT IS LOW TRUE.  
-- SERDAT_N IS INVERTED.  
-- STOP BIT IS FALSE I.E. HIGH.  
-- MINIMUM MESSAGE LENGTH is 1 I.E. MINIMUM SERIAL STREAM LENGTH = 17 BITS.  
--  
-- The implemented function codes are:  
-- CODE 000 = ADC CONFIGURE  
-- CODE 001 = CDS CONFIGURE  
-- CODE 010 = DAC CONFIGURE  
-- CODE 011 = TEL CONFIGURE  
-- CODE 100 = AUX CONFIGURE  
--  
-- Data stream generation is triggered by WRITE_SD.  
--  
-- IMPLEMENTATION NOTES:  
--  
-- ADC Configuration uses:  
-- SERIAL FUNCT CODE "000"  
-- SERIAL DEVADDR RANGE 0 => 3.  
-- SERIAL MSG LENGTH IS 11.  
--  
-- CDS Configuration uses:  
-- SERIAL FUNCT CODE "001".  
-- SERIAL DEVADDR RANGE 0 => 3.  
-- SERIAL MSG LENGTH IS 2.  
--  
-- DAC Configuration uses:  
-- SERIAL FUNCT CODE "010".  
-- SERIAL DEVADDR RANGE 0 => 3.  
-- SERIAL MSG LENGTH IS 16.  
-- | ADDR | FILL | DATA VALUE |  
-- | 2 bits | 2 bits | 12 bits |  
-- MSBIT FIRST.  
--  
-- TEL Configuration uses:  
-- SERIAL FUNCT CODE "011"  
-- SERIAL DEVADDR RANGE 0 => 1.  
-- SERIAL MSG LENGTH IS 8.  
-- |START| ADDR | CFG DATA |  
-- |1 bit| 3 bits | 4 bits |  
-- MSBIT FIRST.  
--  
-- AUX Configuration uses:  
-- SERIAL FUNCT CODE "100".  
-- SERIAL DEVADDR RANGE NOT USED.  
-- SERIAL MSG LENGTH IS 9.  
--  
-- BIT | Function  
-----  
-- 8 | DAC_CLSEL 2  
-- 7 | DAC_CLSEL 1  
-- 6 | DAC_CLSEL 0  
-- 5 | DAC_CLR 1  
-- 4 | DAC_CLR 0  
-- 3 | LED ON  
-- 2 | TEL_SHDN  
-- 1 | BIAS_EN  
-- 0 | CDS_DRIVER_EN  
--  
-- MSBIT FIRST.
```

## cfg\_ser\_out\_v20



```

ADDRESS : in std_logic_vector(7 downto 0);
CFG_DONE : out std_logic;
CFG_SEL : in std_logic_vector(2 downto 0);
DATA : in std_logic_vector(15 downto 0);
DATA_LOCAL : in std_logic_vector(3 downto 0);
RESET : in std_logic;
SERCLK : in std_logic;
WAIT_IN : in std_logic;
WRITE_SD : in std_logic;
SCLK : out std_logic_vector(2 downto 0);
SDATA_OUT : out std_logic;

constant STREAM_BASE_LENGTH: integer := 16;
constant MAX_STREAM_LENGTH: integer := 32;
type statetype is (Idle, Pending, Load, Tx_data);
signal state : statetype;
signal OpReg : std_logic_vector((MAX_STREAM_LENGTH - 1) downto 0);
signal OpRegInit : std_logic_vector((MAX_STREAM_LENGTH - 1) downto 0);
signal OpReg_base : std_logic_vector((STREAM_BASE_LENGTH - 1) downto 0);
signal srl_counter : std_logic_vector(5 downto 0);
signal funct_code : std_logic_vector(2 downto 0);
signal msg_len : std_logic_vector(7 downto 0);
signal stream_length : integer;
signal dac_chan : std_logic_vector(1 downto 0);
signal tel_chan : std_logic_vector(2 downto 0);
signal dev_addr : std_logic_vector(5 downto 0);
signal sclk_msk : std_logic_vector(2 downto 0); -- dlh

SCLK(0)      <= SERCLK and sclk_msk(0);
  
```

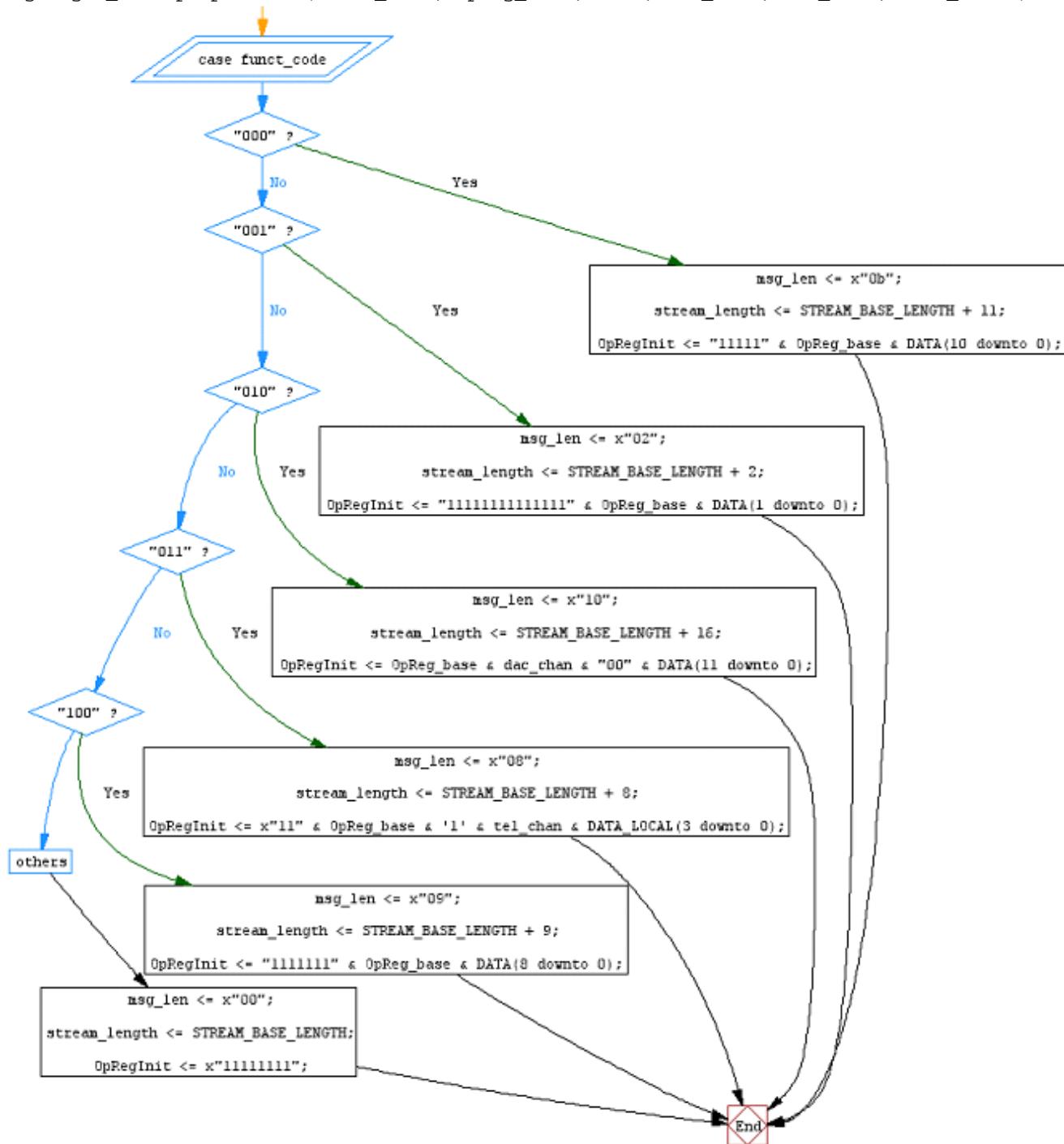
4/5/2006

```

SCLK(1)      <= SERCLK and sclk_msk(1);
SCLK(2)      <= SERCLK and sclk_msk(2); -- dlh need to understand this
funct_code <= CFG_SEL;
OpReg_base <= '0' & funct_code & dev_addr & msg_len(5 downto 0); -- 16 BITS

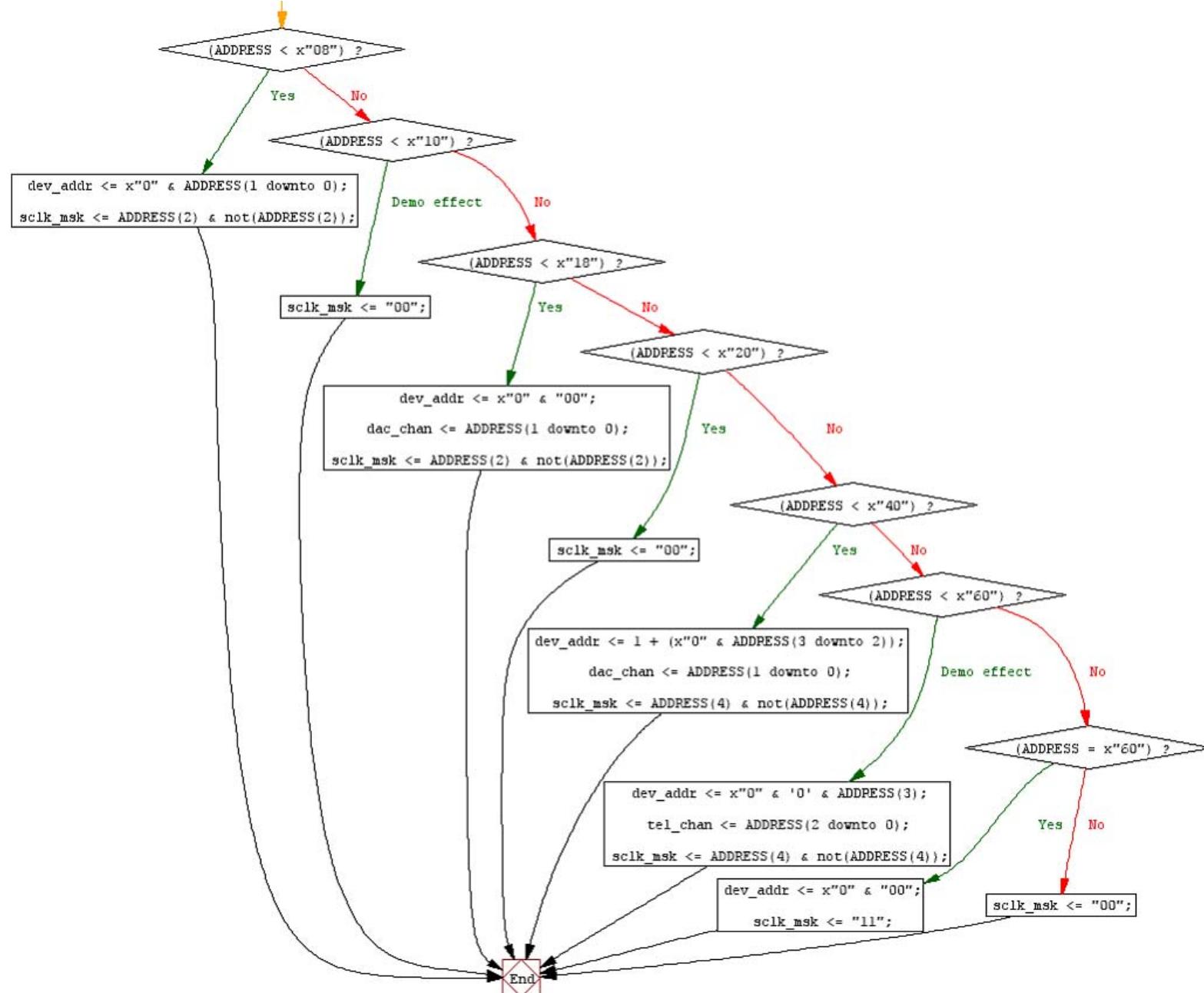
MsgLength_lookup: process (funct_code, OpReg_base, DATA, dac_chan, tel Chan, DATA_LOCAL)

```



4/5/2006

Addr\_Sclk\_lookup: process (ADDRESS)



4/5/2006

FSM: process (SERCLK, RESET)

